EXHIBIT 020

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
10. Method for exchanging messages in an integrated circuit comprising a plurality of modules,	Without conceding that the preamble of claim 10 of the '449 Patent is limiting, Samsung Electronics Co., Ltd.'s (hereinafter, "Samsung") Exynos 1280 system on chip (hereinafter, the "Exynos SoC") is an integrated circuit and performs a method for exchanging messages in an integrated circuit comprising a plurality of modules, either literally or under the doctrine of equivalents.

[1] The Exynos SoC is charted as a representative product made used, sold, offered for sale, and/or imported by or on behalf of Samsung. The citations to evidence contained herein are illustrative and should not be understood to be limiting. The right is expressly reserved to rely upon additional or different evidence, or to rely on additional citations to the evidence cited already cited herein.

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

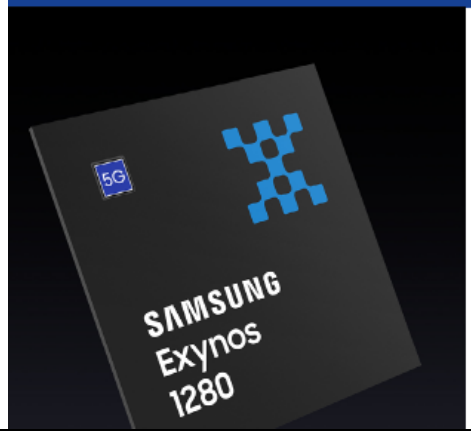
SAMSUNG

Product brief
Create infinite possibilities

Exynos 1280

Highlights

A mobile processor ready for 5G and AI
Advanced ISP and MFC for rich multimedia experience
Powerful octa-core CPU and GPU



5G for all

Exynos 1280 is a mobile processor based on a 64-bit RISC processor. It contains a 5G modem, which is compliant with two types of 5G network (Sub-6GHz and mmWave), as well as all legacy networks. It is built using an advanced 5nm EUV process for high power efficiency.

All-in-one processor for 5G

The Exynos 1280 embedded modem supports both sub-6GHz (Frequency Range

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

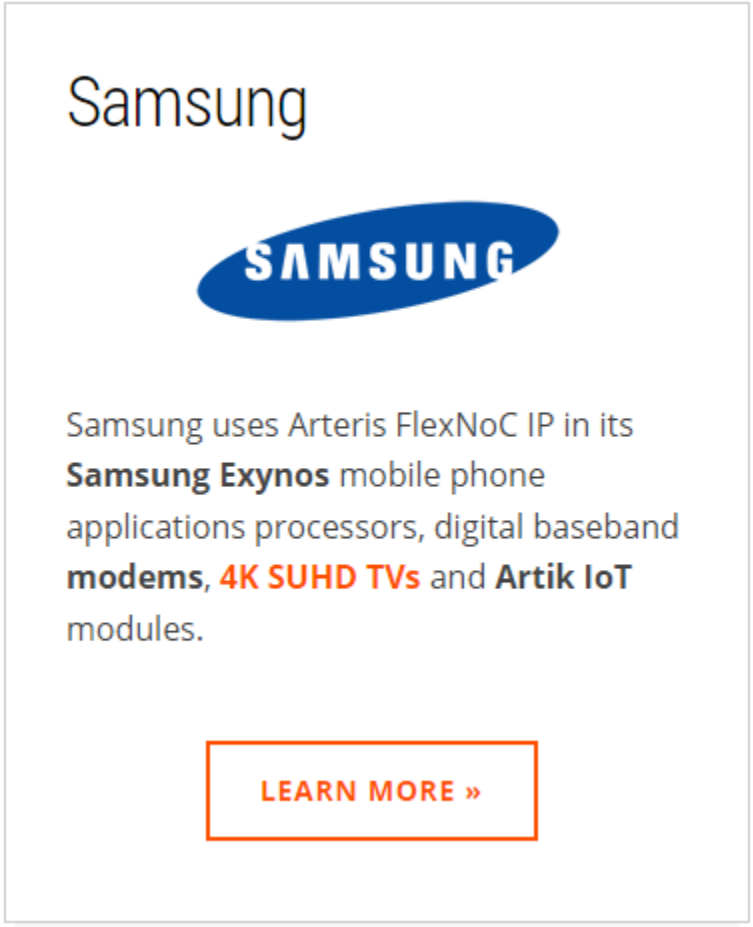
https://semiconductor.samsung.com/resources/brochure/Exynos1280.pdf

The Exynos SoC comprises a plurality of modules, for example Arm Cortex-A78 core, Cortex-A55 core, Arm Mali-G68 GPU, and AI Engine with NPU:

Specifications

	Exynos 1280
CPU	Cortex®-A78 x 2 + Cortex®-A55 x 6
GPU	Mali™-G68
AI	AI Engine with NPU
Modem	5G NR Sub-6GHz 2.55Gbps (DL) / 1.28Gbps (UL) 5G NR mmWave 1.84Gbps (DL) / 0.92Gbps (UL) LTE Cat.18 6CC 1.2Gbps (DL) / Cat.18 2CC 200Mbps (UL)
Connectivity	WiFi 802.11ac MIMO with Dual-band (2.4/5G), Bluetooth® 5.2, FM Radio Rx
GNSS	Quad-constellation multi-signal for L1 and L5 GNSS
Camera	Up to 108MP in single camera mode, Single-camera 32MP @30fps
Video	4K 30fps encoding and decoding
Display	Full HD+@120Hz
Memory	LPDDR4x
Storage	UFS v2.2
Process	5nm

https://semiconductor.samsung.com/resources/brochure/Exynos1280.pdf

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	The Exynos SoC utilizes Arteris network on chip interconnect technology, and/or a derivative thereof, (collectively, the "Arteris NoC") to exchange messages:  https://web.archive.org/web/20210514110614/https://www.arteris.com/customers

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

Arteris IP FlexNoC® Interconnect Licensed by Samsung's System LSI Business for Digital TV Chips

by **Kurt Shuler**, on April 23, 2019

CAMPBELL, Calif. –April 23, 2019– Arteris IP, the world's leading supplier of innovative, silicon-proven network-on-chip (NoC) interconnect semiconductor intellectual property, today announced that Samsung's System LSI Business has renewed multiple Arteris IP FlexNoC Interconnect licenses for use in multiple high-performance digital TV (DTV) processing chips utilizing Samsung's latest semiconductor technology process nodes.

> *Over many years, FlexNoC interconnect IP has helped us accelerate implementation of our digital TV chip designs on our latest semiconductor process nodes. This core interconnect technology is required to develop complex and highly optimized chips in a predictable, low-risk fashion."*

SAMSUNG

*Jaeyoul Lee, Vice President, **Samsung Electronics***

Samsung first licensed FlexNoC interconnect IP in 2010. Since then, Samsung has used Arteris interconnect IP to enable complex SoC architectures in chips like the Exynos mobile processors and other electronic systems.

https://www.arteris.com/press-releases/samsung-lsi-dtv-arteris-ip-flexnoc

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

Arteris Interconnect IP Solution Selected by Samsung for Mobile SoC Deployment

by **Kurt Shuler**, on November 02, 2010

Network-on-Chip (NoC) interconnect technology leader enables higher performance and more cost effective designs for mobile phone systems-on-chip (SoCs)

SUNNYVALE, California — November 2, 2010 — Arteris, Inc., a leading supplier of on-chip interconnect IP solutions, today announced that Samsung Electronics Co., Ltd., has selected Arteris' interconnect solutions for multiple chips within Samsung's mobile SOC products. Samsung chose Arteris interconnect IP to support the high speed inter-chip communication requirements in next generation mobile SOC products.

> *The Arteris interconnect IP offers us a convenient solution to handle the high speed communication needed between our SoC and external modem IC. Our customers will benefit from the lower BOM cost and power consumption as a result of this IP. We look forward to Arteris' interconnect IP helping us shorten development schedules and lower risks associated with compatibility.*

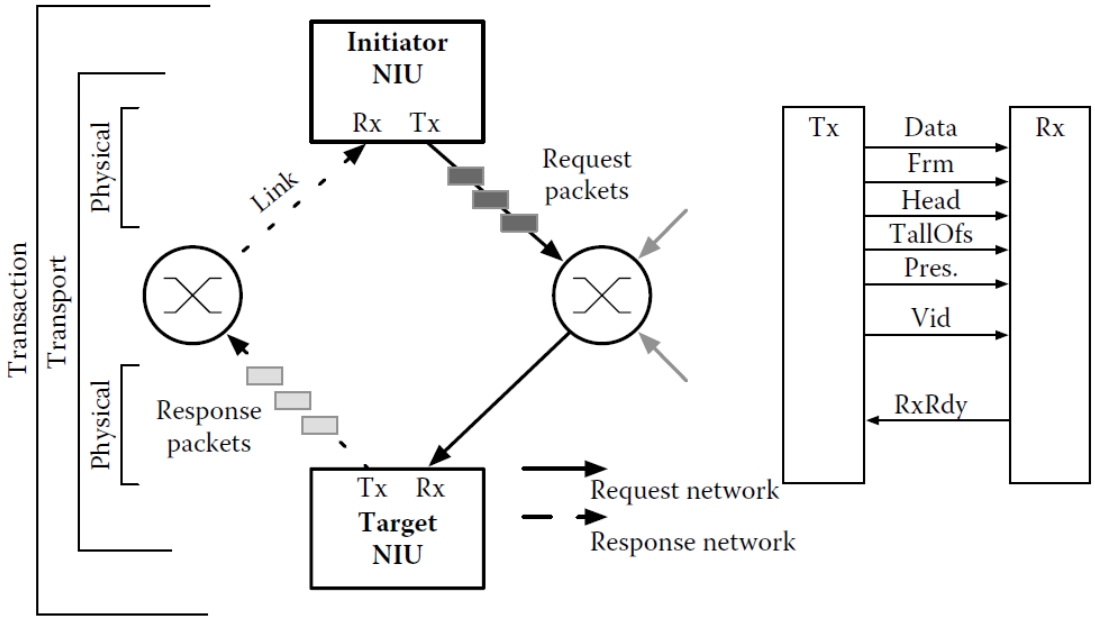
SAMSUNG

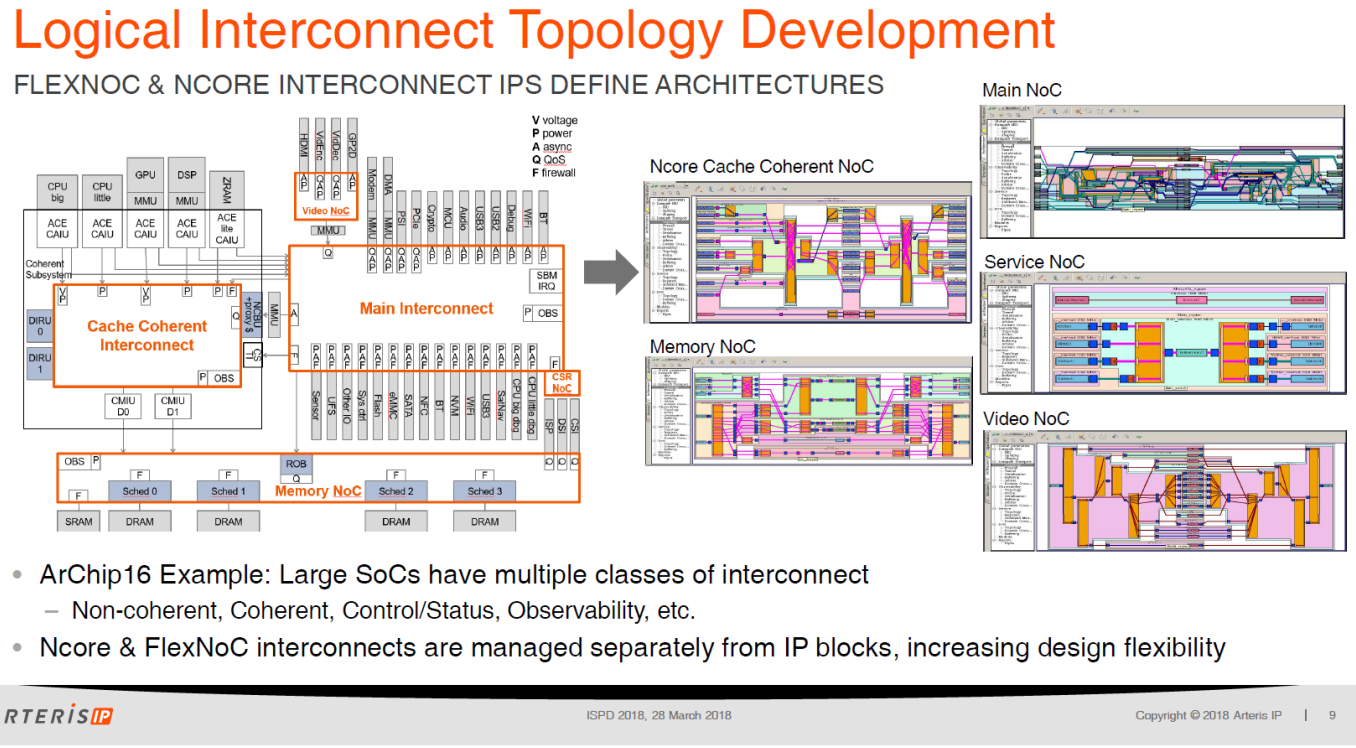
Thomas Kim, Vice President, SoC Platform Development, System LSI, ***Samsung Electronics***

https://www.arteris.com/press-releases/pr_2010_nov_02?hsLang=en-us

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	The Arteris NoC exchanges messages in the Exynos SoC. For example, in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets": ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313; *see id* at 308 (explaining that Chapter 11 of this book describes the function of the Arteris NoC: "In this chapter we will present an MPSoC platform […] using Arteris NoC as communication infrastructure.").
the messages between the modules being	Without conceding that the preamble of claim 10 of the '449 Patent is limiting, the Arteris NoC exchanges messages between modules in the Exynos SoC over connections via a network, wherein said connections comprises a set of communication channels each having a set of

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
exchanged over connections via a network, wherein said connections comprises a set of communication channels each having a set of connection properties, any communication channel being independently configurable,	connection properties any communication channel being independently configurable, either literally or under the doctrine of equivalents. A large SoC, such as the Exynos SoC may include multiple classes of Arteris NoC interconnect network:  *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 9.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	The Exynos SoC utilizes the Arteris NoC to exchange messages over connections via a network, wherein said connections comprises a set of communication channels that are independently configurable. For example, in the the Arteris NoC, "[a]n NTTP transaction is typically made of request packets, traveling through the request network between the master and the slave NIUs, and response packets that are exchanged between a slave NIU and a master NIU through the response network…. Transactions are handed off to the transport layer, which is responsible for delivering packets between endpoints of the NoC (using links, routers, muxes, rated adapters, FIFOs, etc.). Between NoC components, packets are physically transported as cells across various interfaces, a cell being a basic data unit being transported. This is illustrated in Figure 11.1, with one master and one slave node, and one router in the request and response path."

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

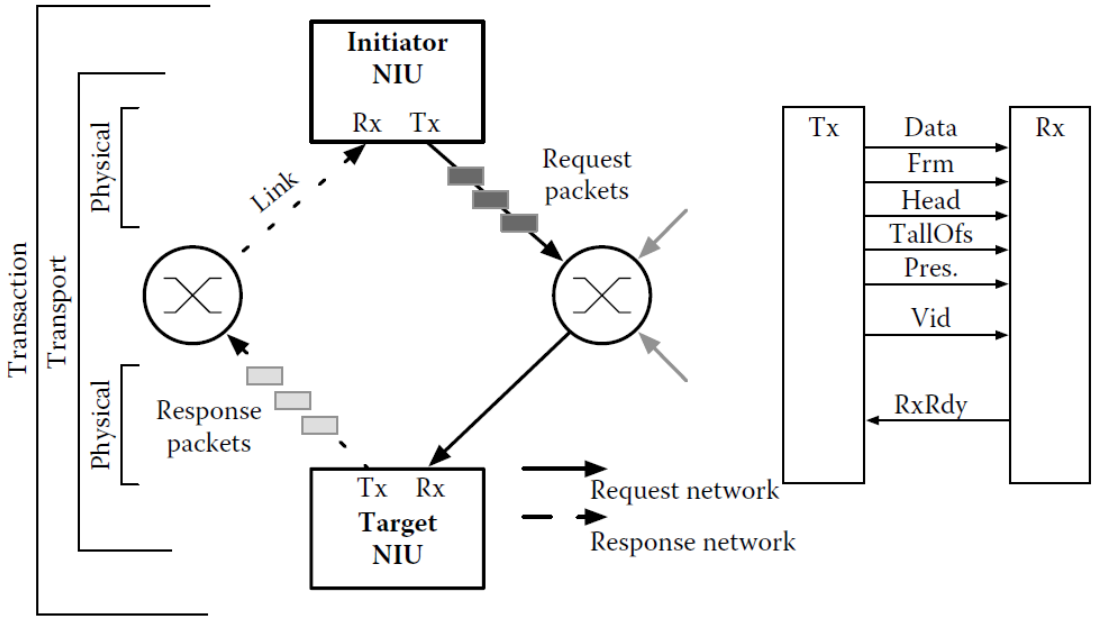
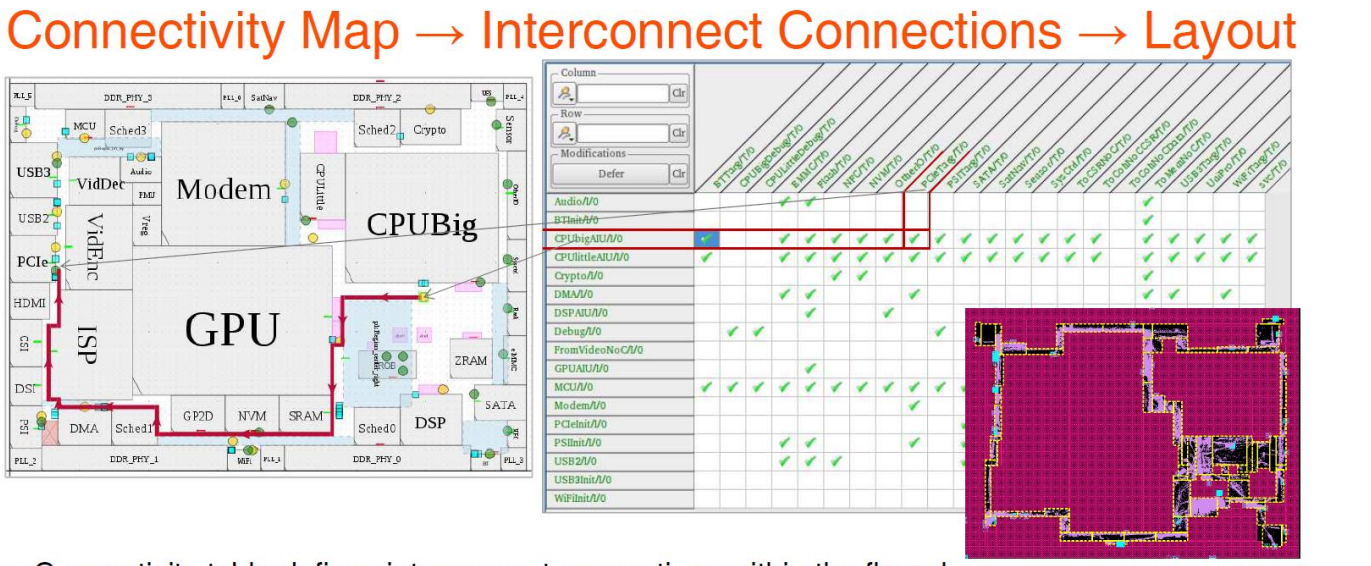


FIGURE 11.1
NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313.

Connections within the Arteris NoC network may be defined by a connectivity table:

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]



- Connectivity table defines interconnect connections within the floorplan
- Routes must pass through available channels in the floorplan
- Connectivity passes from initiator NIU to switch, to link, to RC buffers and finally to target NIU

ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 12

See Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 12.

In the Arteris NoC, "[t]he delivery of packets within the NoC is the responsibility of the physical layer [where the] link size, or width (i.e., number of wires), is set by the designer at design time[and] [o]ne link (represented in Figure 11.1) defines the following signals… Pres.—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2) [and] RxRdy—flow control":

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	11.3.1.3 Physical Layer The delivery of packets within the NoC is the responsibility of the physical layer. Packets, which have been split by the transport layer into cells, are delivered as words that are sent along links. Within a single clock cycle, the physical layer may carry words comprising a fraction of a cell, a single cell, or multiple cells. The link size, or width (i.e., number of wires), is set by the designer at design time and determines the number of cells of one word. NTTP defines five possible link-widths: quarter (QRT), half (HLF), single (SGL), double (DBL), and quad (QUAD). A single-width (SGL) link transmits one cell per clock cycle, a double-width link transmits two cells per clock cycle, and so on. Words travel within point-to-point links, which are independent from other protocol layers: a word is sent through a transmit port, Tx, over a link to a receive port, Rx. The actual number of wires in a link depends on the

maximum cell-width (header, necker, and data cell) and the link-width. One link (represented in Figure 11.1) defines the following signals:

- **Data**—Data word of the width specified at design-time.
- **Frm**—When asserted high, indicates that a packet is being transmitted.
- **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only.
- **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only.
- **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2).
- **Vld**—Data valid: when asserted high, indicates that a word is being transmitted.
- **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy.

This signal set, which constitutes the Media Independent NoC Interface (MINI), is the foundation for NTTP communications.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313-314. The Exynos SoC utilizes the Arteris NoC's connections that comprise a set of communication channels each having a set of connection properties, any communication channel being independently configurable. For example, as noted above, in the Arteris NoC, "[o]ne link (represented in Figure 11.1) defines the following signals… Pres.—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service) [and] RxRdy—flow control." In the Arteris NoC implements Quality of Service (QoS) to "provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic": **Quality of Service (QoS).** The QoS is a very important feature in the interconnect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in Æthereal NoC [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT. In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair. The Arteris NoC supports the following four different traffic classes:

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	• **Real time and low latency (RTLL)**—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. • **Guaranteed throughput (GT)**—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. • **Guaranteed bandwidth (GBW)**—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. • **Best effort (BE)**—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

* Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic.

Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 315-316.

As a further illustration, the Arteris NoC "addresses … varied QoS needs in many ways," including "Dynamic Packet Priorities" and "Dynamic Pressure Propagation":

Arbitration: Dynamic Packet Priorities & Dynamic Pressure Propagation

Arteris Network on Chip technology addresses these varied QoS needs in many ways: First, the interconnect assigns priorities to transactions to ensure they arrive at the target in the proper order to meet system requirements. Priority levels can be attached to individual packets or to all transactions pending on a socket. The interconnect can also assign Dynamic Packet Priorities at runtime.

Second, the interconnect can sense when high priority packets may be blocked or slowed due to downstream traffic congestion and can then clear a path for these high priority packets. This technology, called Dynamic Pressure Propagation, is analogous to a fire truck racing down city streets: All traffic pulls to the side of the road to let the fire truck through.

https://www.arteris.com/end-to-end-quality-of-service-qos

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	As a further illustration, "QoS information may be generated from within the [Arteris] NoC interconnect using Arteris' QoS Generator": Bandwidth Limiters and Rate Regulators Many times architects will want to implement QoS within their SoC but the QoS prioritization data is not available from the individual IP blocks. In this case, QoS information may be generated from within the NoC interconnect using Arteris' QoS Generator. The QoS Generator can instantiate sophisticated, and software programmable, means to regulate interconnect QoS, including: › Bandwidth Limiters – Bandwidth limiters cause a socket to stop accepting requests when a run-time programmable throughput threshold has been exceeded. › Rate Regulators – Rate regulators cause a socket's transactions to be demoted when a bandwidth threshold is reached. This can be considered a smoother version of the bandwidth limiter because transactions are only demoted instead of stalled. https://www.arteris.com/end-to-end-quality-of-service-qos As a further illustration, the Arteris NoC uses "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level": Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16. For the other traffic, "the configuration can be done in architecture":

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	• Best effort traffic can be left untouched. • Latency sensitive traffic may have its urgency modulated as a function of the transaction: *Normal* for writes and *important* for reads. • Soft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives: *Critical* until a specified bandwidth is obtained on a sliding 4 microsecond window, and *normal* thereafter. These settings are set through configuration registers and may be modified while the interconnect is running. The mechanism is called a bandwidth regulator. • On the real-time modem data port, the hurry is fixed at a critical level. *Id.* at 18. As a further illustration, connections within the Arteris NoC may be classified by traffic class and traffic classes may be mapped onto the Arteris interconnect topology:

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

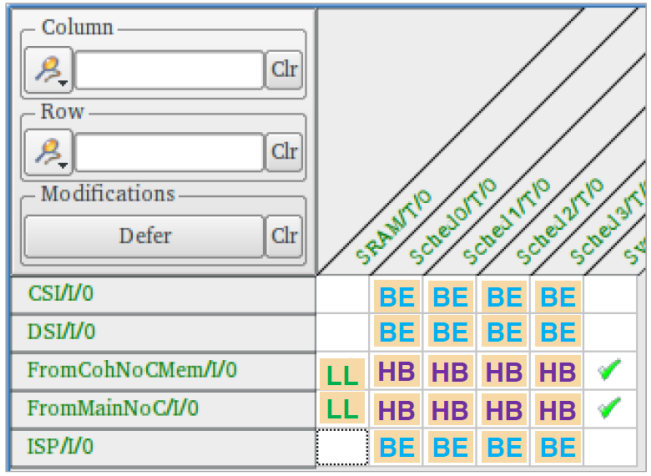
'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

Memory NoC:

Interconnect Topology – Traffic Classes

Classify your IP connections per class of traffic:

Best Effort (**BE**)	Image system
Low Latency (**LL**)	SRAM
High Bandwidth (**HB**)	Main/Coherency

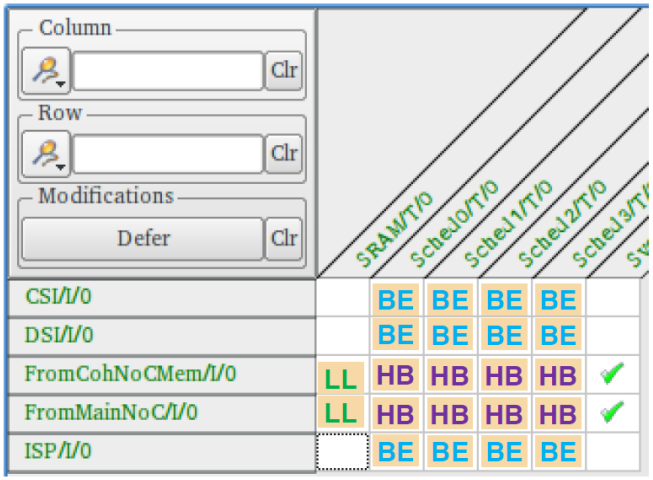


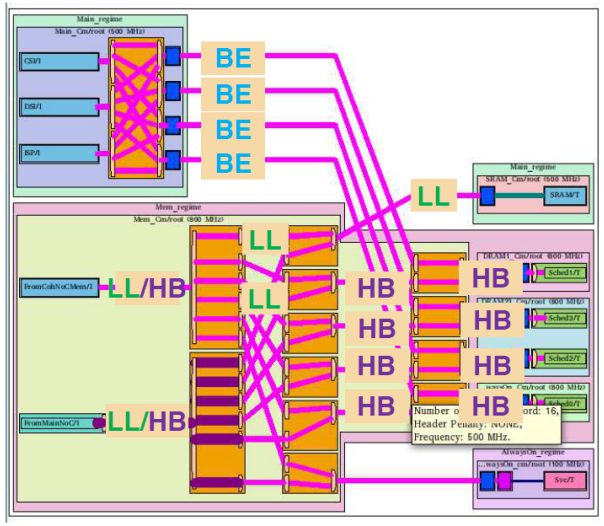
U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

Memory NoC:

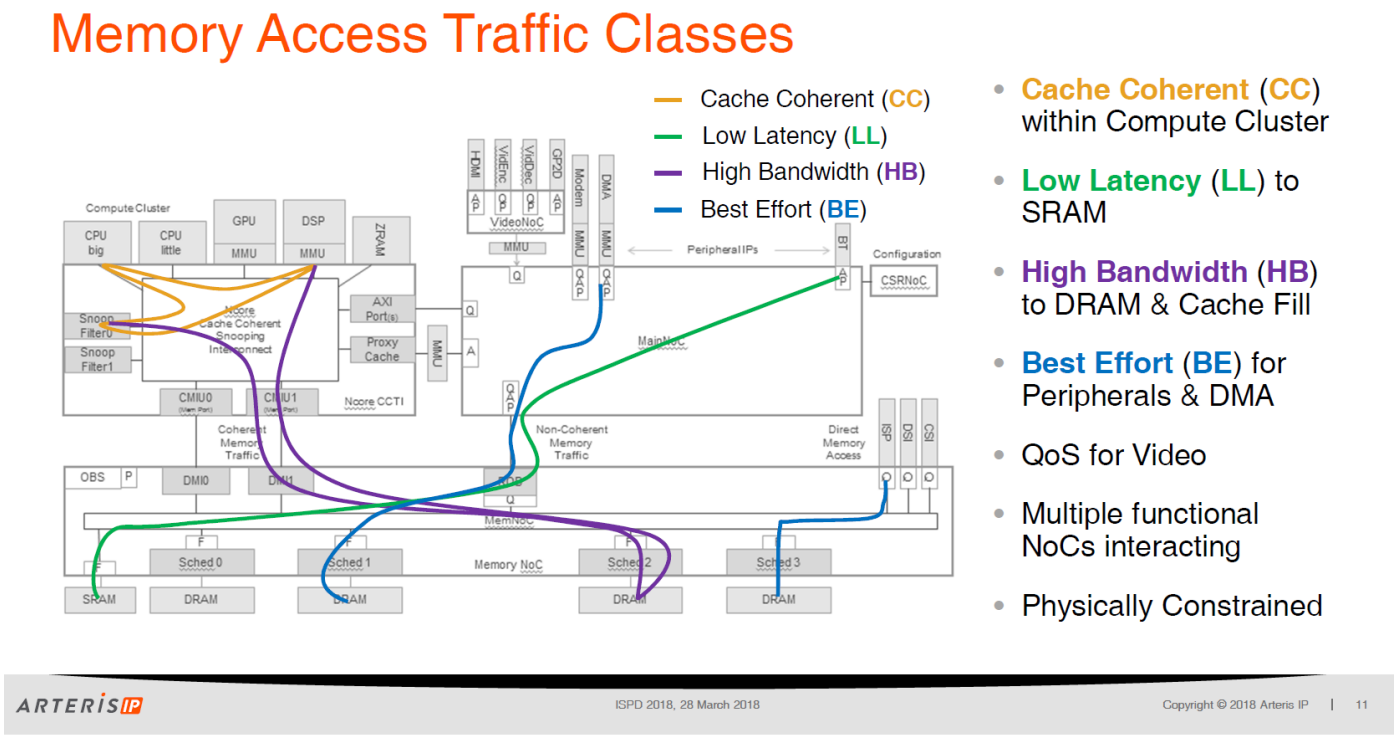
Traffic classes are mapped onto logical interconnect topology





U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]



See Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slides 11, 13, 16.

As a further illustration, in the Arteris NoC, "QoS is supported in the switch using pressure information generated by the IP itself and embedded in NTTP packets" and "[s]ome features [of the switch] can be software-controlled at runtime through the service network":

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	11.3.3.1 Switching The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

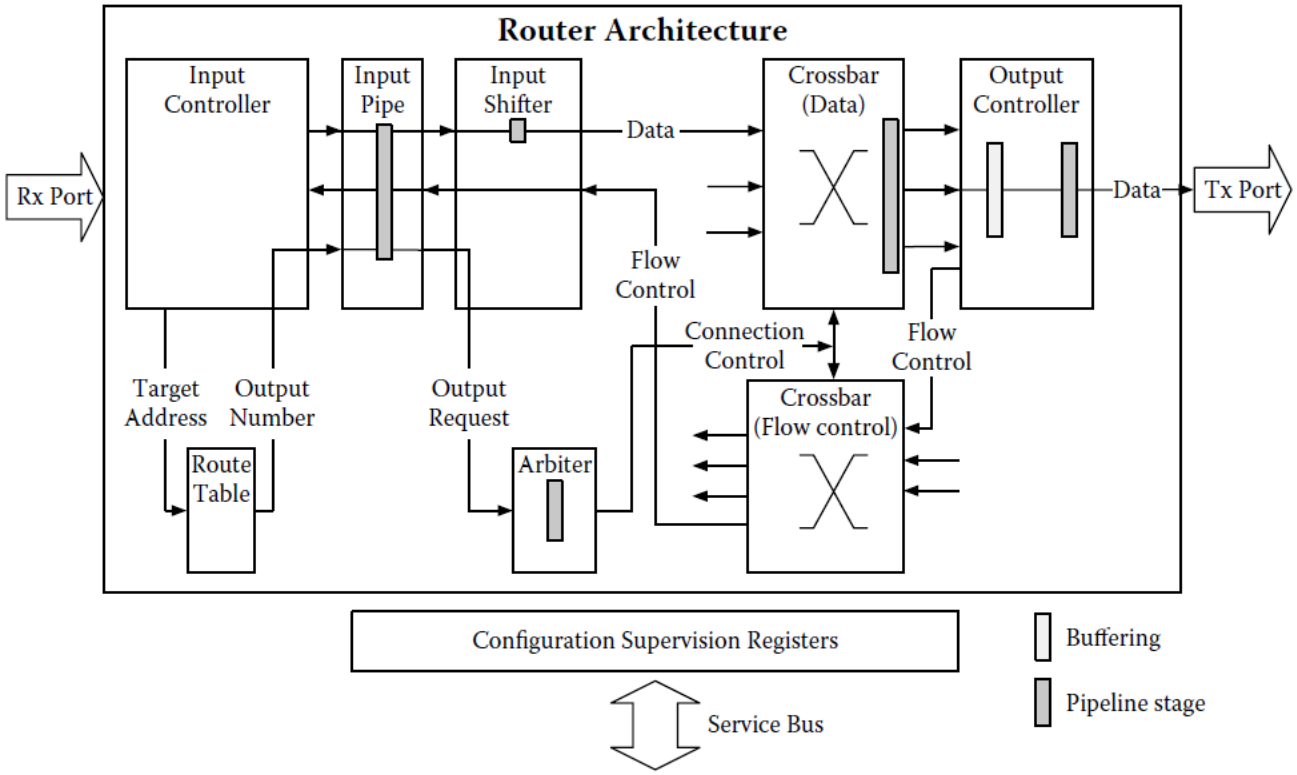
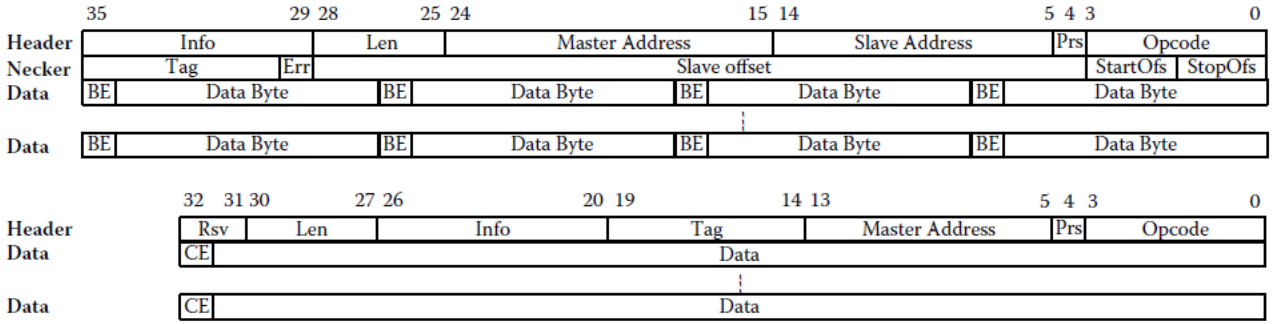


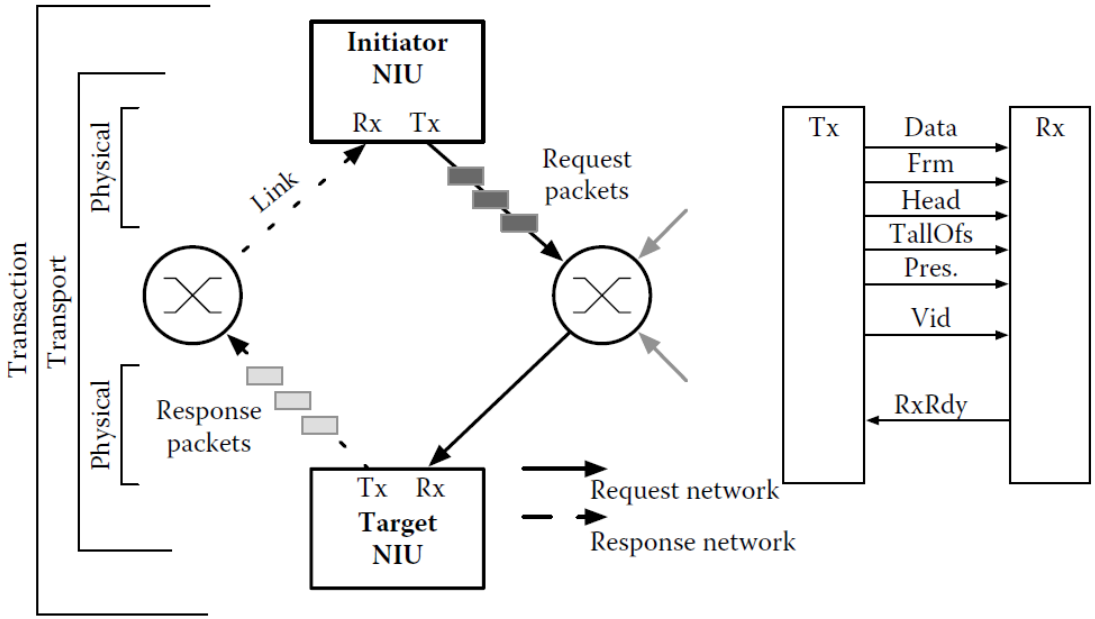
FIGURE 11.6
Packet transportation unit: Router architecture.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	As a further illustration, the "Pres." signal in the NTTP packet "[i]ndicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2)."  **FIGURE 11.2** NTTP packet structure. *See id.* at 313, 314. As a further illustration, in the Arteris NoC, "the routing tables actually used in the switch are parameterizable for each input port of the switch. It is thus possible to use different routing tables for each switch input. Routing tables can optionally be programmed via the service network interface; in this case, their configuration registers appear in the switch register address map." *See id.* at 322.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
wherein said connection through the network supports transactions comprising at least one of outgoing messages from the first module to the second module and return messages from the second module to the first module	Without conceding that the preamble of claim 10 of the '449 Patent is limiting, the Arteris NoC in the Exynos SoC has connections through the network that support transactions comprising at least one of outgoing messages from the first module to the second module and return messages from the second module to the first module, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by the Exynos SoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets": ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313.
and further comprising the steps of: the first module issuing a request	In the Arteris NoC in the Exynos SoC, the first module issues a request for a connection with the second module to a communication manager, wherein the request comprises desired connection properties associated with the sets of communication channels, either literally or under the doctrine of equivalents.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
for a connection with the second module to a communication manager, wherein the request comprises desired connection properties associated with the sets of communication channels;	The first module of the Exynos SoC utilizes the Arteris NoC to issue a request for a connection with the second module to a communication manager. For example, in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets": ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

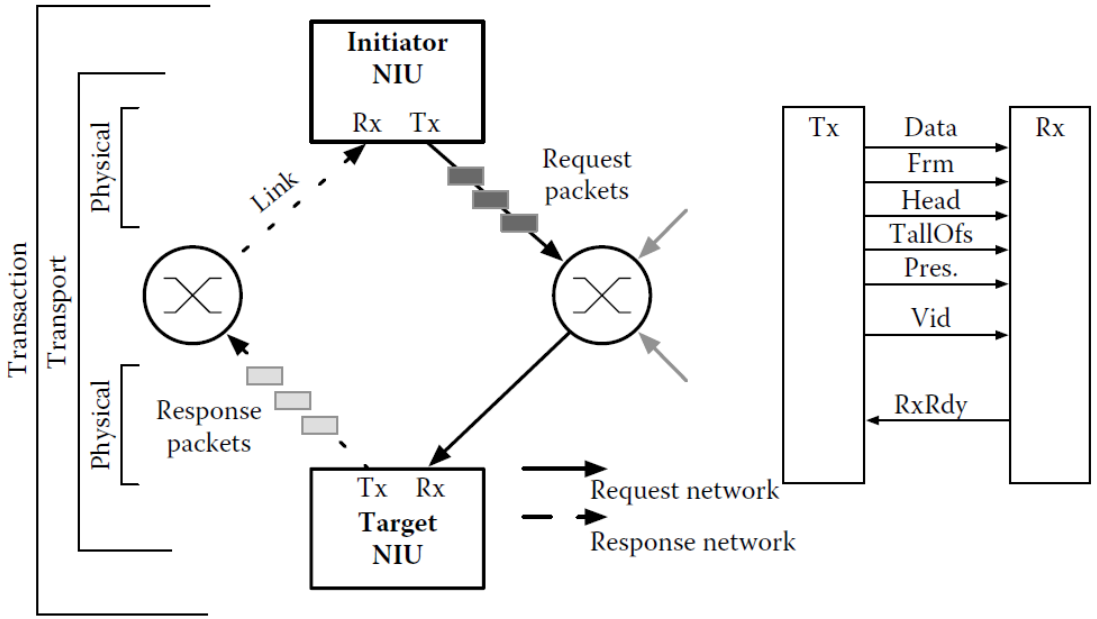


FIGURE 11.1
NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313.

The request issued by first module of the Exynos SoC comprises desired connection properties associated with the sets of communication channels.

For example, in the Arteris NoC, "[t]he delivery of packets within the NoC is the responsibility of the physical layer [where the] link size, or width (i.e., number of wires), is set by the designer at design time[and] [o]ne link (represented in Figure 11.1) defines the following signals… Pres.—

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2) [and] RxRdy—flow control": ***11.3.1.3 Physical Layer*** The delivery of packets within the NoC is the responsibility of the physical layer. Packets, which have been split by the transport layer into cells, are delivered as words that are sent along links. Within a single clock cycle, the physical layer may carry words comprising a fraction of a cell, a single cell, or multiple cells. The link size, or width (i.e., number of wires), is set by the designer at design time and determines the number of cells of one word. NTTP defines five possible link-widths: quarter (QRT), half (HLF), single (SGL), double (DBL), and quad (QUAD). A single-width (SGL) link transmits one cell per clock cycle, a double-width link transmits two cells per clock cycle, and so on. Words travel within point-to-point links, which are independent from other protocol layers: a word is sent through a transmit port, Tx, over a link to a receive port, Rx. The actual number of wires in a link depends on the

maximum cell-width (header, necker, and data cell) and the link-width. One link (represented in Figure 11.1) defines the following signals:

- **Data**—Data word of the width specified at design-time.
- **Frm**—When asserted high, indicates that a packet is being transmitted.
- **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only.
- **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only.
- **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2).
- **Vld**—Data valid: when asserted high, indicates that a word is being transmitted.
- **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy.

This signal set, which constitutes the Media Independent NoC Interface (MINI), is the foundation for NTTP communications.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313-314. As a further example, in the Arteris NoC, "QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition" where the "pressure signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed" and the "pressure information will be embedded in the NTTP packet at the NIU level": **Quality of Service (QoS).** The QoS is a very important feature in the interconnect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in Æthereal NoC [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT. In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair. The Arteris NoC supports the following four different traffic classes:

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	• **Real time and low latency (RTLL)**—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. • **Guaranteed throughput (GT)**—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. • **Guaranteed bandwidth (GBW)**—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. • **Best effort (BE)**—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

* Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic.

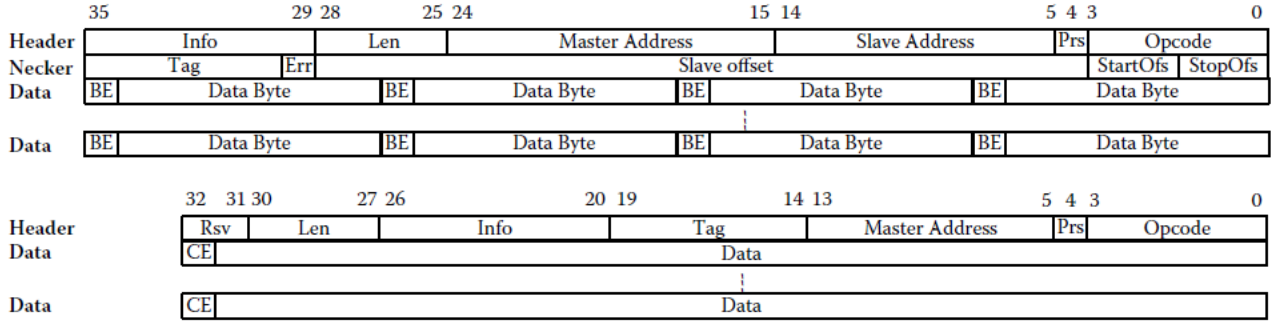


FIGURE 11.2
NTTP packet structure.

Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313, 315-316.

As a further example, in the Arteris NoC, "QoS is supported in the switch using pressure information generated by the IP itself and embedded in NTTP packets" and the router architecture includes blocks such as "Input Controller," "Flow Control," "Crossbar (Flow control)" "Route Table" and "Arbiter":

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	11.3.3.1 Switching The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

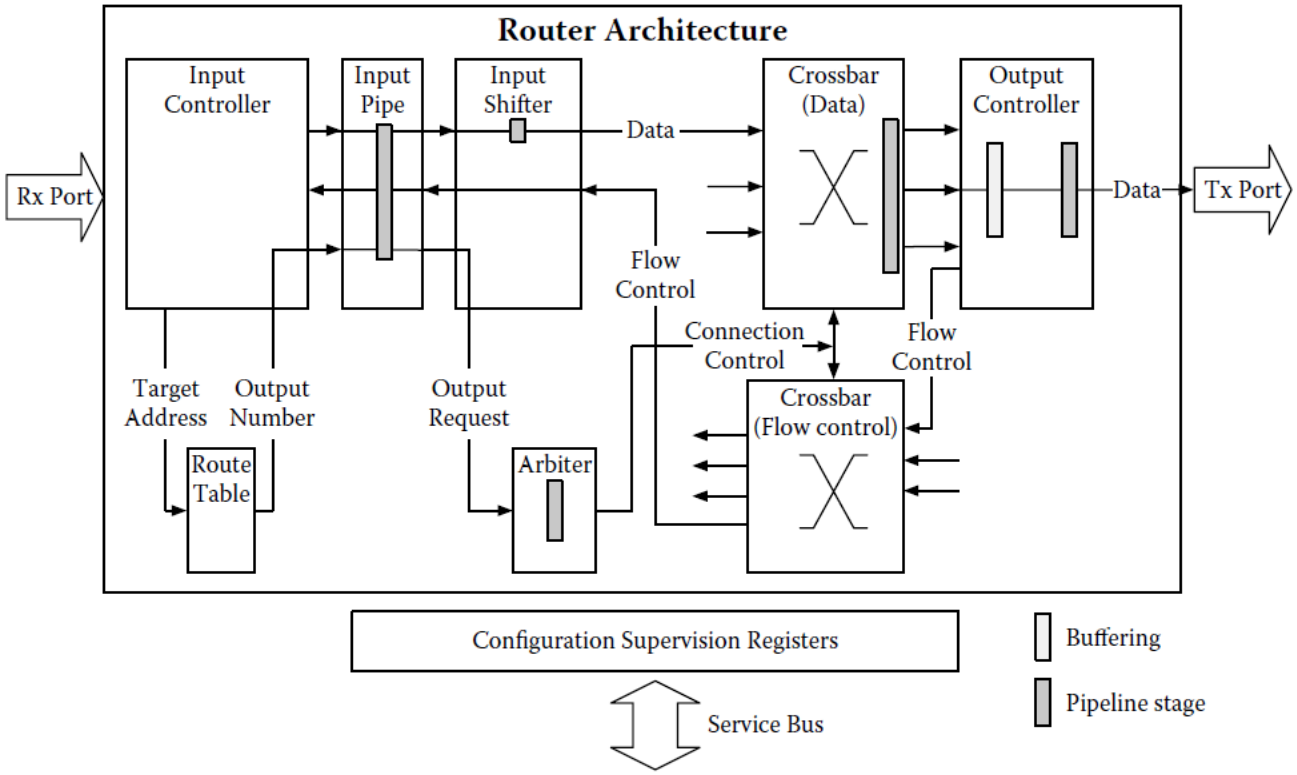


FIGURE 11.6
Packet transportation unit: Router architecture.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
the communication manager forwarding the request to a resource manager; the resource manager determining whether a target connection with the desired connection properties is available;	In the Arteris NoC in the Exynos SoC, the communication manager forwards the request to a resource manager and the resource manager determines whether a target connection with the desired connection properties is available, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by Exynos SoC, "QoS is supported in the switch" which "choos[es] the route" using a "routing table"; "arbitrat[es]"; and "switch[es]" and the router architecture includes blocks such as "Input Controller," "Flow Control," "Crossbar (Flow control)" "Route Table" and "Arbiter": ***11.3.3.1 Switching*** The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

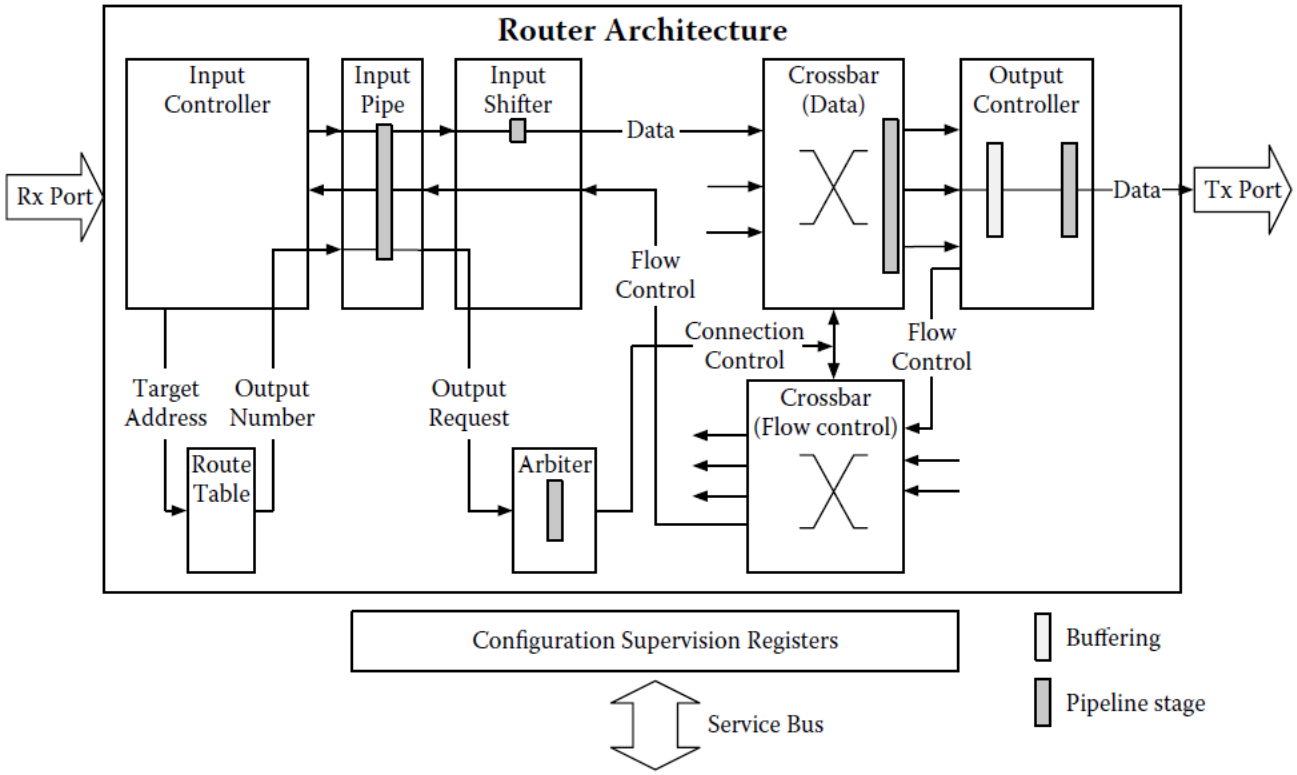


FIGURE 11.6
Packet transportation unit: Router architecture.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	As a further illustration, in the Arteris NoC, "the pressure information used to define the preferred traffic class (QoS) of the requesting inputs… [t]he pressure information is given top priority by the switch arbiter" and "the input controller extracts pertinent data from packet headers, forwards it to the routing table, fetches back the target output number, and then sends a request to the arbiter. After arbitration is granted, the input controller transmits the rest of the packet to the crossbar. The request to the arbiter is sustained as long as the last word of the packet has not been transferred. Upon transferring the last cell of the packet, the arbiter is allowed to select a new input." *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 321, 322.
the resource manager responding with the availability of the target connection to the communication manager; and	In the Arteris NoC in the Exynos SoC, the resource manager responds with the availability of the target connection to the communication manager, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by the Exynos SoC, "the pressure information used to define the preferred traffic class (QoS) of the requesting inputs… [t]he pressure information is given top priority by the switch arbiter" and "the input controller extracts pertinent data from packet headers, forwards it to the routing table, fetches back the target output number, and then sends a request to the arbiter. After arbitration is granted, the input controller transmits the rest of the packet to the crossbar. The request to the arbiter is sustained as long as the last word of the packet has not been transferred. Upon transferring the last cell of the packet, the arbiter is allowed to select a new input." *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 321, 322.

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	As a further illustration, in the Arteris NoC, "[t]he arbiter ensures that the connection matrix (a row per input and a column per output) contains at most one connection per column, that is, a given output is not fed by two inputs at the same time. The dual guarantee — at most one connection per row — is handled by the input controller. Each output has an arbiter that includes prefiltering. For maximum flexibility, each port can specify its own arbiter from the list of available arbiters (random, round robin, LRU, FIFO, or fixed priority)." *Id.* at 322-323.
the target connection between the first and second module being established based on the available properties of said communication channels of said connection.	In the Arteris NoC in the Exynos SoC, the target connection between the first and second module is established based on the available properties of said communication channels of said connection, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by the Exynos SoC, "QoS is supported in the switch" which "choos[es] the route" using a "routing table"; "arbitrat[es]"; and "switch[es]":

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	11.3.3.1 Switching The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]

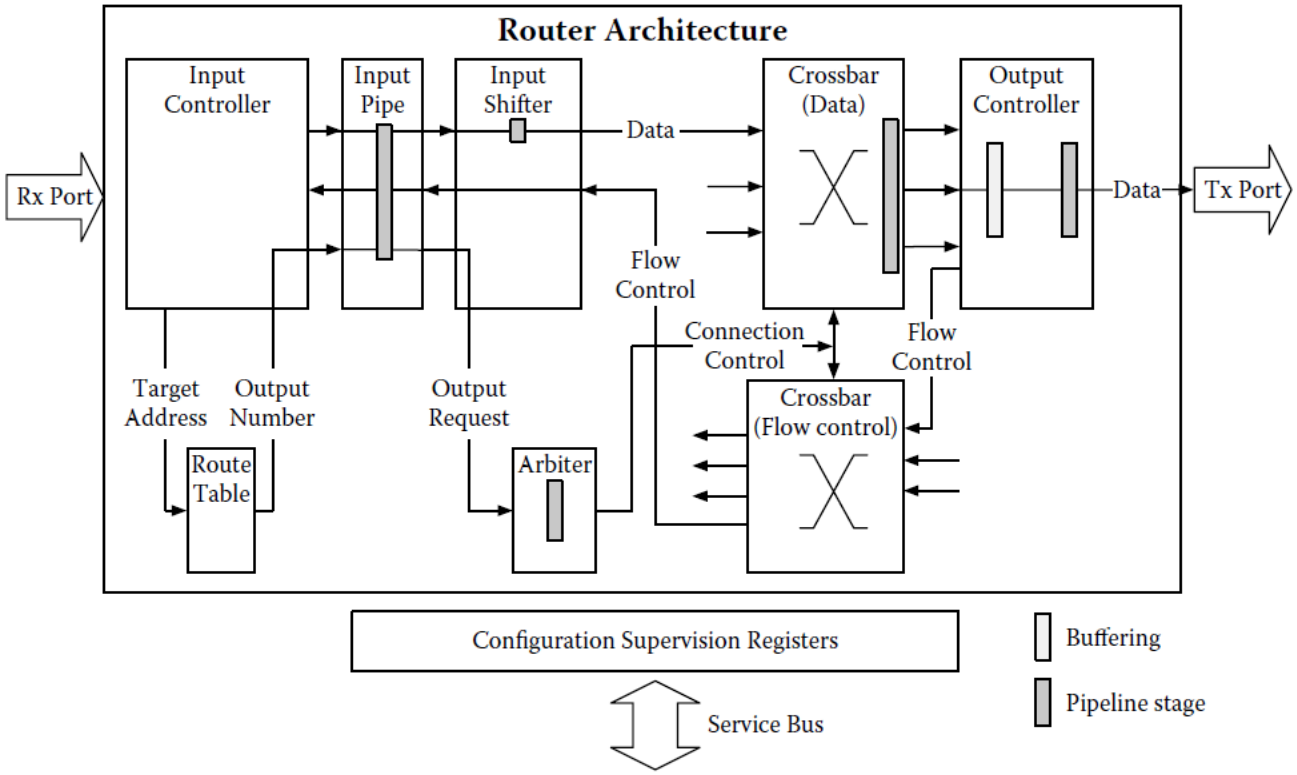


FIGURE 11.6
Packet transportation unit: Router architecture.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

'449 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	In the Arteris NoC, "the pressure information used to define the preferred traffic class (QoS) of the requesting inputs… [t]he pressure information is given top priority by the switch arbiter" and "the input controller extracts pertinent data from packet headers, forwards it to the routing table, fetches back the target output number, and then sends a request to the arbiter. After arbitration is granted, the input controller transmits the rest of the packet to the crossbar. The request to the arbiter is sustained as long as the last word of the packet has not been transferred. Upon transferring the last cell of the packet, the arbiter is allowed to select a new input." *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 321, 322. As a further illustration, in the Arteris NoC, "[t]he arbiter ensures that the connection matrix (a row per input and a column per output) contains at most one connection per column, that is, a given output is not fed by two inputs at the same time. The dual guarantee— at most one connection per row—is handled by the input controller. Each output has an arbiter that includes prefiltering. For maximum flexibility, each port can specify its own arbiter from the list of available arbiters (random, round robin, LRU, FIFO, or fixed priority)." *Id.* at 322-323. As a further illustration, in the Arteris NoC, "[t]he crossbar implements datapath connection between inputs and outputs. It uses the connection matrix produced by the arbiter to determine which connections must be established. It is equivalent to a set of *m* muxes (one per output port), each having *n* inputs (one per input port)." *Id.* at 323.